

Penyelesaian Algoritma *Pattern Generation* dengan Model *Arc-Flow* pada *Cutting Stock Problem* (CSP) Satu Dimensi

Putra BJ Bangun, Sisca Octarina, Rika Apriani
Jurusan Matematika Fakultas MIPA Universitas Sriwijaya
Indralaya, Indonesia
e-mail: teger4959@ymail.com

Abstract—Permasalahan optimasi dalam kasus pengkombinasian pola pemotongan yang hanya memperhatikan salah satu sisi pemotongan dikenal dengan *Cutting Stock Problem* (CSP) satu dimensi. Penelitian ini menggunakan algoritma *pattern Generation* dan model *arc-flow* untuk menyelesaikan permasalahan pola pemotongan kayu. Berdasarkan hasil dan pembahasan didapatkan bahwa algoritma *pattern generation* menghasilkan pola-pola pemotongan yang optimal tanpa adanya *trim loss*. Pola-pola tersebut selanjutnya dimodelkan ke dalam model *arc-Flow*. Model yang terbentuk hanya menggunakan kendala pemenuhan permintaan dan kendala non negatif, sedangkan kendala yang berkaitan dengan konservasi *flow* tidak digunakan.

Keywords—*cutting stock problem; algoritma pattern generation; model arc-flow; trim loss*

I. PENDAHULUAN

Permasalahan yang sering dijumpai dalam bidang perindustrian seperti industri kayu, kertas, baja, dan *fiber* adalah meminimumkan bahan baku. Bahan baku dengan ukuran yang lebih besar disebut *stock sheets*, umumnya harus dipotong-potong menjadi bentuk yang lebih kecil yang disebut *item*. Setiap *item* mempunyai jumlah permintaan tertentu, sehingga pola pemotongan sangat diperlukan agar sisa bahan baku dapat seminimal mungkin [6].

Masalah pemotongan bahan baku dalam dunia optimasi dikenal dengan sebutan *Cutting Stock Problem* (CSP) dan sisa pemotongan selanjutnya dikenal dengan istilah *trim loss*. *Trim loss* adalah sisa pemotongan berlebih yang disebabkan oleh peletakan pola pemotongan yang kurang tepat sehingga mengakibatkan ketidakefisienan penggunaan bahan baku [6].

CSP mempunyai 3 jenis yaitu CSP satu dimensi, CSP dua dimensi, dan CSP tiga dimensi. Pendimensian CSP dibedakan berdasarkan sisi *trim loss*. Penelitian ini membahas CSP satu

dimensi pada bidang industri kayu yang hanya melihat permasalahan pada sisi panjang.

CSP satu dimensi pernah dibahas oleh beberapa peneliti sebelumnya dengan menggunakan beberapa teknik dan material yang berbeda. *Column Generation Technique* (CGT) lebih tepat digunakan dalam penyelesaian CSP satu dimensi dibandingkan dengan algoritma Balas yang dimodifikasi karena tidak semua solusi yang dihasilkan berupa solusi biner [8]. Metode lain yang dapat digunakan dalam menyelesaikan CSP adalah model *arc-flow* dengan kendala sisi. Model ini memiliki seperangkat konservasi kendala *flow* dan satu kumpulan kendala permintaan untuk memastikan bahwa permintaan dari setiap *item* terpenuhi [6]. Selanjutnya, formulasi *arc-flow* dengan kompresi grafik untuk menyelesaikan CSP dengan pola biner (0-1) telah dilakukan [4].

Selain itu, algoritma *pattern generation* telah diperkenalkan oleh [9] untuk menyelesaikan masalah pemotongan bahan. Algoritma tersebut dilakukan untuk mengurangi jumlah pemotongan yang tidak layak. Selain itu, prosedur pola sederhana dikembangkan untuk menyelesaikan masalah tambahan. Kelemahan algoritma ini yaitu mengandalkan pohon pencarian dalam pencarian pola pemotongan. Ketika dihadapkan pada kasus yang menghasilkan pola pemotongan yang banyak, pohon pencarian tidaklah selalu efektif. Model *arc-flow* dengan kendala permintaan diharapkan dapat mengoptimalkan pencarian pola optimal dan meminimalkan *trim loss*.

Berdasarkan latar belakang tersebut, penelitian ini membahas bagaimana penyelesaian algoritma *pattern generation* dengan model *arc-flow* pada CSP satu dimensi. Model ini diuji sebagai CSP satu dimensi material kayu pada permasalahan yang ada di salah satu depot di Indralaya.

Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

http://ars.ilkom.unsri.ac.id

II. METODE PENELITIAN

Penelitian ini meneliti tentang pola pemotongan kayu. Data yang diuji berasal dari salah satu depot kayu di Indralaya. Algoritma *pattern generation* diterapkan untuk menentukan pola pemotongan yang optimal. Diagram pohon dibuat dari cabang atas ke bawah dan dari kiri ke kanan. Tabel *trim loss* dibuat berdasarkan gambar pencabangan yang telah dibuat. Data-data tersebut digunakan untuk membuat model *arc-flow* yang selanjutnya diselesaikan menggunakan bantuan program LINDO.

III. ALGORITMA PATTERN GENERATION

Pemotongan pada CSP biasanya merupakan ukuran bahan baku dengan lebar standar w dibawa ke n ukuran dengan lebar dan panjang tertentu dari w dan l masing-masing. Bahan baku dengan lebar standar w_k ($k = 1, 2, \dots, h$) dipotong ke n ukuran dengan lebar w_i dan panjang l_i ($i = 1, 2, \dots, n$). Tidak ada batasan pada panjang potongan standar. Untuk kepentingan tertentu, panjang potongan dapat digabungkan untuk menghasilkan panjang yang diinginkan. Tujuannya untuk menentukan pola pemotongan yang dapat meminimumkan *trim loss* dengan memenuhi permintaan. Model CSP dapat dilihat pada model (1).

Fungsi tujuan :

Minimumkan

$$z = \sum_{k=1}^h \sum_{j=1}^{m_k} c_{jk} x_{jk} + \sum_{i=1}^n w_i s_i \quad (1)$$

dengan kendala

$$\sum_{k=1}^h \sum_{j=1}^{m_k} a_{ijk} x_{jk} - s_i = l_i \quad \text{untuk semua } i \quad (2)$$

$$x_{jk}, s_i, c_{jk}, a_{ijk} \geq 0 \quad \text{untuk semua } i, j \text{ dan } k \quad (3)$$

dengan :

a_{ijk} adalah jumlah unit dengan lebar w_i yang akan dipotong menurut pola ke j dari potongan ke k ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$; $k = 1, 2, \dots, h$).

x_{jk} adalah panjang potongan ke k yang akan dipotong menurut pola ke j .

c_{jk} adalah sisa pemotongan dari potongan ke k yang akan dipotong menurut pola ke j .

s_i adalah panjang yang berlebih/kelebihan panjang yang akan menghasilkan potongan dengan lebar ke w_i .

m_k adalah jumlah pola pemotongan yang dapat menghasilkan potongan ke k .

Pembangkitan pola pemotongan yang fisibel diperoleh melalui sebuah pohon pencarian. Level dari pohon menyatakan lebar yang diperlukan disusun dengan urutan menurun dimana ukuran terbesar di level pertama sedangkan ukuran terkecil diletakkan di level terendah dari pohon. Verteks awal dari level I menyatakan lebar standar dari potongan ke k (w'_k) yang digunakan untuk membangkitkan pola. Oleh karena itu, pohon pencarian terpisah digunakan untuk membangkitkan pola tergantung dengan masing-masing lebar standar.

Cabang dari level i pada pohon pencarian menyatakan perkalian antara jumlah unit dengan lebar i yang dipotong menurut pola ke j dengan lebar ke w_i . Perkalian ini menyatakan jumlah lebar yang dipotong dari potongan ke k untuk memenuhi lebar ke w_i . Verteks awal pada level ke i menyatakan sisa lebar setelah memenuhi pemotongan tertentu dari cabang $i - 1$ sebelumnya. Verteks akhir pada level tertinggi dari pohon menunjukkan sisa pemotongan yang dihasilkan dari pola pemotongan yang berbeda.

Misalkan $w_1 > w_2 > w_3 > \dots > w_n$. Untuk setiap gulungan ke k , sebuah matriks $[a_{ijk}]$ dengan elemen $n \times m_k$ harus diketahui.

Untuk elemen pertama di kolom pertama :

$$a_{11k} = \left\lfloor \frac{w'_k}{w_1} \right\rfloor \quad (4)$$

Untuk elemen ke dua di kolom pertama :

$$a_{21k} = \left\lfloor \frac{w'_k - a_{21k} w_1}{w_2} \right\rfloor \quad (5)$$

Untuk elemen ke i di kolom pertama :

$$a_{i1k} = \left\lfloor \frac{(w'_k - \sum_{z=1}^{i-1} a_{z1k} w_z)}{w_i} \right\rfloor \quad (6)$$

Untuk elemen ke i di kolom ke j , dirumuskan :

$$a_{ijk} = \left\lfloor \frac{(w'_k - \sum_{z=1}^{i-1} a_{zjk} w_z)}{w_i} \right\rfloor \quad (7)$$

Trim Loss pola pemotongan dapat ditentukan oleh :

$$c_{jk} = w'_k - \sum_{i=1}^n a_{ijk} w_i \quad (8)$$

Secara terperinci langkah-langkah pada algoritma *pattern generation* diuraikan sebagai berikut :

1. Susun ukuran lebar yang diperlukan w_i ($i = 1, 2, \dots, n$) dengan urutan menurun.
2. Gunakan Persamaan (7) untuk mengisi kolom pertama ($j = 1$) dari matriks.
3. Gunakan Persamaan (8) untuk mencari sisa pemotongan yang dihasilkan dari pola pemotongan.
4. Atur level indeks (indeks baris) i ke $n - 1$.
5. Periksa verteks sekarang pada level i , misal verteks (i, j) . Jika verteks memiliki nilai sama dengan nol ($a_{ijk} = 0$), lanjut ke Langkah 7. Jika tidak bangkitkan kolom baru $j = j + 1$ dengan elemen-elemen berikut :
 - $a_{zjk} = a_{zi-1k}$ ($z = 1, \dots, i - 1$) elemen untuk mengisi verteks yang mendahului verteks sekarang i, j .
 - $a_{ijk} = a_{ij-1k} - 1$ elemen untuk mengisi verteks sekarang i, j .
 - Isi sisa verteks dari kolom j . Misalnya $a_{i+1jk}, a_{i+2jk}, \dots, a_{njk}$ dengan Persamaan (7).
6. Gunakan Persamaan (8) untuk mencari *trim loss* yang dihasilkan dari pola pemotongan ke j . Kembali ke Langkah 4.
7. Pengurangan i , misal $i = i - 1$. Jika $i > 0$ ulangi ke Langkah 5. Jika tidak, berhenti.

Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

http://ars.ilkom.unsri.ac.id

IV. MODEL ARC FLOW

Model *arc-flow* yang bertujuan meminimumkan masalah *flow* (aliran) satu dimensi CSP dikenalkan [6]. Setiap pemotongan sesuai dengan jalan asiklik yang digambarkan oleh graf $G = (V, A)$, dengan $V = \{0, 1, \dots, W\}$ sebagai himpunan dari $W+1$ verteks yang menentukan posisi dari *stock sheet*, dan $A = \{(u, v) : 0 \leq u < v \leq W \text{ dan } u - v = w_i, \forall i = 1, \dots, m\}$ sebagai himpunan dari busur. Model tersebut dirumuskan sebagai permasalahan minimum *flow*, dengan variabel f_{uvi} bersesuaian dengan busur (u, v) , misalnya jumlah dari lebar *item* $v - u$ ditempatkan pada jarak dari u unit dari awal *stock sheet* yang diberikan, dan variabel z sesuai dengan total *flow* yang bergerak sepanjang graf dan dapat dilihat sebagai aliran (*flow*) balik dari verteks W ke verteks 0. Formulasi umum *arc-flow* adalah sebagai berikut :

Maksimumkan/minimumkan z (9)

dengan kendala

$$\sum_{(u,v,i) \in A} f_{uvi} - \sum_{(v,u',i) \in A} f_{uv'i} = \begin{cases} -z & \text{jika } v = S, \\ z & \text{jika } v = T, \\ 0 & \text{untuk } v \in V \setminus \{S, T\}, \end{cases} \quad (10)$$

$$\sum_{(u,v,i) \in A} f_{uvi} \geq b_i \quad i \in \{1, \dots, m\}, \quad (11)$$

$$f_{uvi} \geq 0 \text{ dan integer, } \forall (u, v, i) \in A, \quad (12)$$

dengan :

z adalah variabel yang dapat dilihat sebagai busur umpan balik dari verteks T ke S ;

b_i adalah permintaan dari *item* ke- i ;

A adalah himpunan dari busur yang memiliki komponen (u, v, i) ;

f_{uvi} adalah jumlah aliran di sepanjang busur (u, v, i) ;

m adalah jumlah *item* yang berbeda;

S adalah verteks awal; dan

T adalah verteks akhir.

Kendala (10) berkaitan dengan konservasi *flow* dan kendala (11) memastikan bahwa semua permintaan tercukupi.

V. HASIL DAN PEMBAHASAN

Kasus yang digunakan pada penelitian ini adalah pemotongan kayu untuk membuat kusen pintu dan jendela. Setiap kusen terdiri dari beberapa potongan kayu dengan bermacam-macam ukuran. Tabel 1 menjelaskan jumlah potongan kayu yang dibutuhkan untuk membuat setiap jenis pintu dan jendela.

Tabel 1. Nama-Nama Produk dan Ukuran Potongan Kayu yang Dibutuhkan

Nama Produk	Jumlah Potongan Kayu yang Dibutuhkan (Buah)					
	2 m	1,6 m	1 m	0,8 m	0,5 m	0,4 m
Kusen Pintu Jenis ke-1	2	-	1	-	-	-
Kusen Pintu Jenis ke-2	2	-	-	1	-	-
Kusen Jendela Jenis ke-1	-	2	-	-	-	2
Kusen Jendela Jenis ke-2	-	2	-	-	2	-
Kusen Jendela Jenis ke-3	-	3	-	2	-	-
Kusen Pintu Lengkung	2	3	-	-	-	2

Sedangkan data jumlah kusen yang dipesan dapat dilihat pada Tabel 2.

Tabel 2. Jenis-Jenis Kusen dan Jumlah Kusen yang Dipesan

Jenis-Jenis Kusen	Jumlah
Kusen Pintu Jenis ke-1	5
Kusen Pintu Jenis ke-2	15
Kusen Jendela Jenis ke-1	32
Kusen Jendela Jenis ke-2	2
Kusen Jendela Jenis ke-3	2
Kusen Pintu Lengkung	1

Ukuran standar kayu persediaan yang digunakan Depot Kusen memiliki panjang 4 meter, lebar 0,12 meter dan tinggi 0,06 meter. Berdasarkan data pada Tabel 1 dan Tabel 2 dapat dikelompokkan ukuran potongan-potongan kayu yang dibutuhkan untuk membuat kusen pintu dan kusen jendela serta jumlah masing-masing potongan.

Jumlah total potongan kayu didapat dengan cara menghitung potongan yang dibutuhkan untuk membuat semua kusen yang dipesan yang dapat dilihat pada Tabel 3. Jumlah total setiap potongan kayu yang dibutuhkan adalah nilai dari variabel b_i pada setiap kendala yang bersesuaian.

Prosiding
ANNUAL RESEARCH SEMINAR 2016
 6 Desember 2016, Vol 2 No. 1

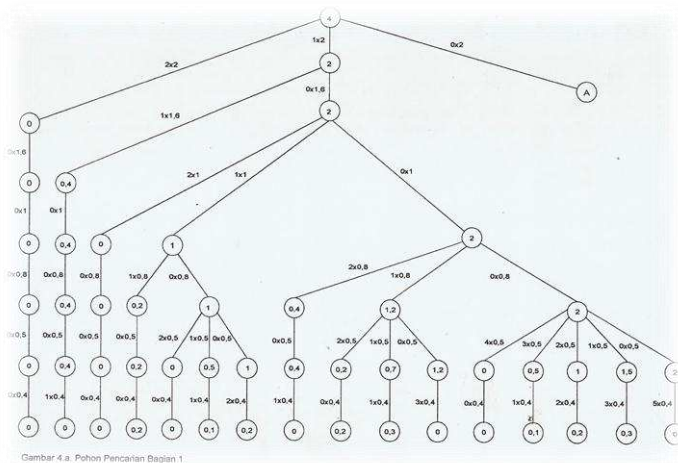
ISBN : 979-587-626-0 | UNSRI

<http://ars.ilkom.unsri.ac.id>

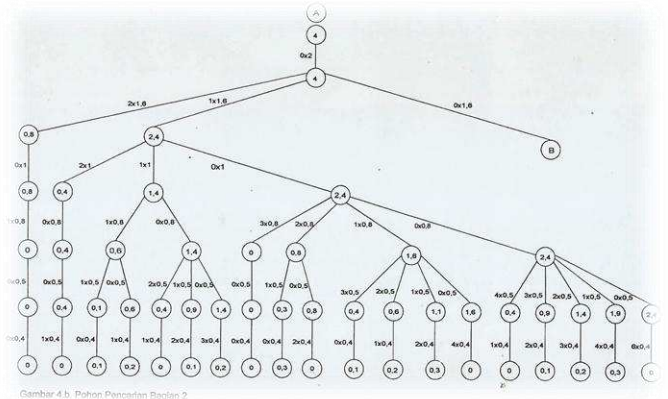
Tabel 3. Data Ukuran Potongan-Potongan Kayu dan Jumlahnya yang Dibutuhkan

No	Jenis-jenis Kusen	Jumlah Kusen yang Dipesan	Jumlah Potongan Kayu yang Dibutuhkan untuk Membuat Kusen yang Dipesan (Buah)					
			2 m	1,6 m	1 m	0,8 m	0,5 m	0,4 m
1	Kusen Pintu Jenis ke-1	5	10	-	5	-	-	-
2	Kusen Pintu Jenis ke-2	15	30	-	-	15	-	-
3	Kusen Jendela Jenis ke-1	32	-	64	-	-	-	64
4	Kusen Jendela Jenis ke-2	2	-	4	-	-	4	-
5	Kusen Jendela Jenis ke-3	2	-	6	-	4	-	-
6	Kusen Lengkung	1	2	3	-	-	-	2
Jumlah			42	77	5	19	4	66

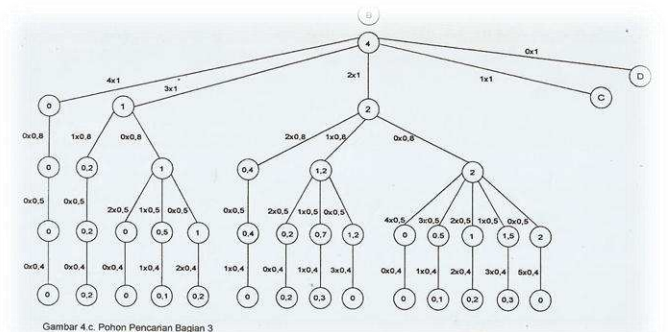
Berdasarkan Tabel 3 terlihat bahwa potongan kayu berukuran 0,4 meter dibutuhkan sebanyak 66 buah. Pohon pencarian digunakan untuk menentukan pola pemotongan yang sesuai dengan panjang dan ukuran yang diminta agar menghasilkan *trim loss* yang minimum. Penerapan algoritma *pattern generation* dalam menentukan pola pemotongan untuk kayu dengan panjang standar 4 meter, dihasilkan pada Gambar 1.



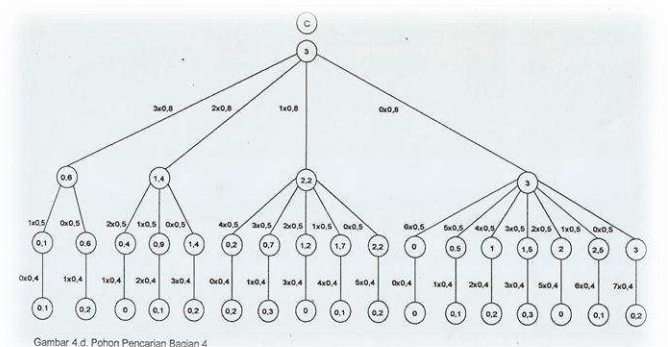
Gambar 1. Pohon Pencarian Bagian 1



Gambar 2. Pohon Pencarian Bagian 2



Gambar 3. Pohon Pencarian Bagian 3

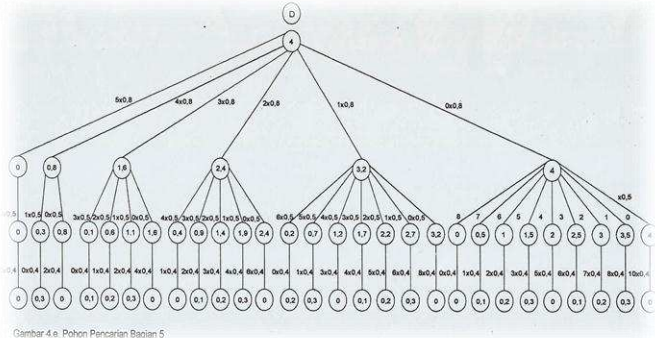


Gambar 4. Pohon Pencarian Bagian 4

Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

http://ars.ilkom.unsri.ac.id



Gambar 5. Pohon Pencarian Bagian 5

Algoritma *pattern generation* yang telah digunakan untuk menentukan pola pemotongan menghasilkan *trim loss* masing-masing yaitu 0; 0,1; 0,2 dan 0,3 meter. Dari 94 pola pemotongan yang diperoleh, dipilih pola pemotongan dengan *trim loss* yang paling minimum yaitu 0 meter seperti pada Tabel 4 dan selanjutnya akan diimplementasikan ke dalam model *arc-flow*.

Tabel 4. Tabel Pola-Pola Pemotongan

Pola ke-j	Jumlah Potongan						Trim Loss (m)
	2 m	1,6 m	1 m	0,8 m	0,5 m	0,4 m	
1	2	0	0	0	0	0	0
2	1	1	0	0	0	1	0
3	1	0	2	0	0	0	0
4	1	0	1	0	2	0	0
5	1	0	0	2	0	1	0
6	1	0	0	1	0	3	0
7	1	0	0	0	4	0	0
8	1	0	0	0	0	5	0
9	0	2	0	1	0	0	0
10	0	1	2	0	0	1	0
11	0	1	1	0	2	1	0
12	0	1	0	3	0	0	0
13	0	1	0	2	0	2	0
14	0	1	0	1	0	4	0
15	0	1	0	0	4	1	0
16	0	1	0	0	0	6	0
17	0	0	4	0	0	0	0
18	0	0	3	0	2	0	0
19	0	0	2	2	0	1	0
20	0	0	2	1	0	3	0
21	0	0	2	0	4	0	0
22	0	0	2	0	0	5	0
23	0	0	1	2	2	1	0
24	0	0	1	1	2	3	0
25	0	0	1	0	6	0	0
26	0	0	1	0	2	5	0
27	0	0	0	5	0	0	0
28	0	0	0	4	0	2	0
29	0	0	0	3	0	4	0
30	0	0	0	2	4	1	0
31	0	0	0	2	0	6	0
32	0	0	0	1	4	3	0
33	0	0	0	1	0	8	0
34	0	0	0	0	8	0	0
35	0	0	0	0	4	5	0
36	0	0	0	0	0	10	0

Tabel 4 menggambarkan bahwa, apabila 1 buah kayu persediaan berukuran standar 4 meter dipotong menggunakan pola pemotongan ke-j, maka dihasilkan sejumlah potongan kayu berukuran K_i , dengan $K_1 = 2$ meter, $K_2 = 1,6$ meter, $K_3 = 1$ meter, $K_4 = 0,8$ meter, $K_5 = 0,5$ meter, dan $K_6 = 0,4$ meter. Data pada tabel pola pemotongan tersebut selanjutnya digunakan untuk membentuk model *arc-flow* seperti pada Model (13) sebagai berikut:

Minimumkan

$$z = 4x_1 + 4x_2 + 4x_3 + 4x_4 + 4x_5 + 4x_6 + 4x_7 + 4x_8 + 4x_9 + 4x_{10} + 4x_{11} + 4x_{12} + 4x_{13} + 4x_{14} + 4x_{15} + 4x_{16} + 4x_{17} + 4x_{18} + 4x_{19} + 4x_{20} + 4x_{21} + 4x_{22} + 4x_{23} + 4x_{24} + 4x_{25} + 4x_{26} + 4x_{27} + 4x_{28} + 4x_{29} + 4x_{30} + 4x_{31} + 4x_{32} + 4x_{33} + 4x_{34} + 4x_{35} + 4x_{36}$$

dengan kendala

$$2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 84$$

$$x_2 + 2x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} \geq 123,2$$

$$2x_3 + x_4 + 2x_{10} + x_{11} + 4x_{17} + 3x_{18} + 2x_{19} + 2x_{20} +$$

$$2x_{21} + 2x_{22} + x_{23} + x_{24} + x_{25} + x_{26} \geq 5$$

$$2x_5 + x_6 + x_9 + 3x_{12} + 2x_{13} + x_{14} + 2x_{19} + x_{20} + 2x_{23} +$$

$$x_{24} + 5x_{27} + 4x_{28} + 3x_{29} + 2x_{30} + 2x_{31} + x_{32} + x_{33} \geq 15,2$$

$$2x_4 + 4x_7 + 2x_{11} + 4x_{15} + 2x_{18} + 4x_{21} + 2x_{23} + 2x_{24} +$$

$$6x_{25} + 2x_{26} + 4x_{30} + 4x_{32} + 8x_{34} + 4x_{35} \geq 2$$

$$x_2 + x_5 + 3x_6 + 5x_8 + x_{10} + x_{11} + 2x_{13} + 4x_{14} + x_{15} +$$

$$6x_{16} + x_{19} + 3x_{20} + 5x_{22} + x_{23} + 3x_{24} + 5x_{26} + 2x_{28} +$$

$$4x_{29} + x_{30} + 6x_{31} + 3x_{32} + 8x_{33} + 5x_{35} + 10x_{36} \geq 26,4$$

$$x_j \geq 0 \text{ dan integer, } j = 1, 2, 3, \dots, 36 \quad (13)$$

Fungsi tujuan z pada Persamaan (13) menyatakan banyak kayu yang akan dipotong untuk membuat kusen pintu dan jendela. Kendala yang digunakan hanya kendala untuk memastikan permintaan pemesanan terpenuhi dan kendala nonnegatif, untuk kendala yang berkaitan dengan konservasi *flow* tidak digunakan karena dalam pemilihan pola pemotongan yang digunakan adalah pola pemotongan dengan *trim loss* yang paling minimum yaitu 0 meter, sehingga tidak ada fungsi yang menyatakan panjang *trim loss*.

Model *arc-flow* yang sudah terbentuk selanjutnya diselesaikan dengan menggunakan metode *Branch and Bound* dan program LINDO 6.1 untuk mendapatkan hasil akhir. Berdasarkan hasil penyelesaian Model (13) diperoleh solusi optimal yaitu $x_2 = 82$, $x_3 = 2$, $x_4 = 1$ dan $x_9 = 21$. Hal ini berarti dalam memenuhi permintaan pemesanan kusen pintu dan kusen jendela dapat digunakan pola pemotongan ke 2 sebanyak 82 kali, pola pemotongan ke 3 sebanyak 2 kali, pola pemotongan ke 4 sebanyak 1 kali dan pola pemotongan ke 9 sebanyak 21 kali. Panjang total kayu yang diperoleh untuk membuat kusen yang dipesan adalah 424 meter artinya jumlah total kayu persediaan panjang 4 meter yang dipotong adalah 106 buah.

Prosiding
ANNUAL RESEARCH SEMINAR 2016
6 Desember 2016, Vol 2 No. 1

ISBN : 979-587-626-0 | UNSRI

<http://ars.ilkom.unsri.ac.id>

VI. KESIMPULAN

Berdasarkan hasil yang telah dicapai, dapat diambil kesimpulan bahwa algoritma *pattern generation* dapat menentukan semua pola pemotongan dengan terperinci hingga menghasilkan *trim loss* 0 meter. Proses pencabangan membutuhkan iterasi yang cukup panjang dan algoritma ini tidak memberikan total *trim loss* yang dihasilkan. Ketika, pola-pola pemotongan yang terbentuk dimodelkan ke dalam model *arc-flow*, total *trim loss* dan jumlah kayu yang dibutuhkan dapat diketahui. Model *arc-flow* yang terbentuk hanya menggunakan kendala pemenuhan permintaan dan kendala nonnegatif, sedangkan kendala yang berkaitan dengan konservasi *flow* tidak digunakan karena dalam pemilihan pola pemotongan yang digunakan adalah pola pemotongan dengan *trim loss* yang paling minimum yaitu 0 meter, sehingga tidak ada fungsi yang menyatakan panjang *trim loss*.

DAFTAR PUSTAKA

- [1] C. Aldridge, S.J. Chapman, R. Gower, R. Leese, C. McDiarmid, M. Shepherd, H. Tuenter, H. Wilson and A. Zinober, "Pattern reduction in paper cutting," in Report of the 29th European Study Group with Industry, University of Oxford, 1996.
- [2] C. Antens, T. Gustaffson, A. Paulshus, C. Stroh, and M. Wiese, "Pattern reduction in the paper cutting problem," in The Ninth ECMI Modelling Week. Penn Sate A Publication Research, 1996.
- [3] D.S. Chen, R.G. Batson, and Y. Dang, Applied Integer Programming Modeling and Simulation. New Jersey: John Wiley & Sons, 2010.
- [4] F. Brandao and J.P. Perdroso, "Cutting stock with binary patterns: arc-flow formulation with graph compression," in Technical Report Series : DCC-2013-09., 2013.
- [5] J. Karehlati, "Solving the cutting stock problem in the steel industry," in Master's Thesis Submitted in Partial Fulfillment of The Requirements for degree of Master of Science in Technology, Helsinki University of Technology, 2002.
- [6] R. Macedo, C. Alves, and J.M.V. Carvalho, "Arc flow model for the two-dimensional guillotine cutting stock problem," in Computers & Operational Research, Volume 37 Issues 6, pp. 991-1001, June 2010.
- [7] R.W. Haessler and P.E. Sweeney, "Cutting stock problems and solution procedures," in European Journal of Operational Research, Vol. 54, pp. 141-150, 1991.
- [8] S. Octarina, P.B.J. Bangun, and D. Setiadi, "Optimasi trim loss pada cutting stock problem menggunakan column generation technique dan algoritma balas yang dikembangkan," in Proceeding Annual Research Seminar, Vol. 1 No. 1, pp. 57-58, 2015.
- [9] S.M.A. Suliman, "Pattern generating procedure for the cutting stock problem," in International Journal of Production Economics, vol. 74, pp. 293-301, 2001.